

bash

1 Variablen

1.1 Wert zuweisen

VARIABLE=WERT (Text oder ganzzahlige Werte)

1.2 Wert auslesen

\$VARIABLE oder \${VARIABLE}

2 Quoting: Maskieren (unwirksam machen) von Sonderzeichen

2.1 Backslash \

maskiert genau ein darauffolgendes Zeichen

2.2 Double Quotes (Anführungszeichen) " "

maskieren *nicht* \ \$ ' ,

2.3 Single Quotes (Hochkomma) ' '

maskieren alles

3 alias

Tastatur-Shortcut, Beispiel: `alias ls='ls --color=auto'`

4 Expansion (Substitution): Ersetzen von Sonderzeichen durch Text

4.1 geschweifte Klammern { , }

Beispiel 1 {rot,gelb,blau}.txt ⇒ rot.txt gelb.txt blau.txt

Beispiel 2 (kartesisches Produkt) {a,b,c}{1,2}.txt ⇒ a1.txt a2.txt b1.txt b2.txt c1.txt c2.txt

Beispiel 3 (Verschachtelung) {a,b{c,d}}e.txt ⇒ ae.txt bce.txt bde.txt

Beispiel 4 `mkdir -p umsatz/200{8,9}/q{1,2,3,4}`

4.2 Tilde ~

wird ersetzt durch den Pfad des Homeverzeichnisses des aktuellen Benutzers, z.B. `/home/normal`

4.3 Variablen \$VARIABLE oder \${VARIABLE}

die geschweiften Klammern werden benötigt, wenn hinter dem Variablennamen kein Leerzeichen steht, z.B. `echo ${NAME}s Katze ist doof`

4.4 Kommandosubstitution \$() oder ` `

Beispiele `verzeichnisliste=$(ls /)` oder `verzeichnisliste=` ls / ``

4.5 Arithmetische Terme $\$(())$ und Rechenanweisungen $(())$

Beispiele `echo $\$((2+3)*4)$` und `((x += 4))`

Operatoren `+ - * / % ** = += -= *= /= ++ --`

4.6 Pathname Expansion (Globbing)

- `*` beliebig viele beliebige Zeichen (außer `/`)
- `?` genau ein beliebiges Zeichen (außer `/`)
- `[xyz]` genau ein Zeichen `x` oder `y` oder `z`
- `[k-p]` genau ein Zeichen `k` oder `l` oder `m`, ..., `p` (**nicht case-sensitive**)
- `[!a]` genau ein Zeichen, aber nicht `a`
- `[!xyz]` genau ein Zeichen, aber nicht `x` oder `y` oder `z`

5 Listen

`Kommando1 ; Kommando2 ; Kommando3` (nacheinander)
`Kommando1 && Kommando2 && Kommando3` (nur bei Erfolg (exit 0))
`Kommando1 || Kommando2 || Kommando3` (nur bei Fehler)

6 Beispiel mit Variablen- und Kommandosubstitution

	Variablen- substitution	Kommando- substitution	Befehlszeile	ausgeführtes Kommando	Ausgabe
			<code>x=beispiel</code>		
Single Quotes	-	-	<code>echo 'touch \$x'</code>		<code>touch \$x</code>
Double Quotes	ja	-	<code>echo "touch \$x"</code>		<code>touch beispiel</code>
Backticks	ja	ja	<code>echo `touch \$x`</code>	<code>touch beispiel</code>	Zeilenumbruch
$\$()$	ja	ja	<code>echo $\\$(touch \$x)$</code>	<code>touch beispiel</code>	Zeilenumbruch